

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims

1-30. Canceled.

31. (New) Method for processing data packets and managing compiled packet filtering code, comprising:

compiling packet filtering code and writing the compiled code to a memory;

receiving data packets for processing;

processing the received packets according to the compiled code; and

managing the compiled code in a plurality of pieces.

32. (New) Method according to claim 31, further comprising:

incrementally compiling at least one rule into at least one piece of compiled code by a rule compiling entity;

transmitting said piece of compiled code from the rule compiling entity to a packet processing entity;

pausing processing of packets by said packet processing entity;

writing said piece of compiled code to the memory; and

restarting processing packets by said packet processing entity.

33. (New) Method according to claim 31, further comprising:

incrementally compiling at least one rule into at least one piece of compiled code by a rule compiling entity;

signaling from said rule compiling entity to a packet processing entity that a new piece of compiled code is compiled;

signaling from said packet processing entity to said rule compiling entity that said packet processing entity is ready for said piece of compiled code to be stored;

writing said piece of compiled code to said memory; and

signaling from said rule compiling entity to said packet processing entity that said piece of compiled code is written to said memory.

34. (New) Method according to claim 31 wherein said pieces are code pages having a predetermined length.

35. (New) Method according to claim 34 wherein shadow paging is used.

36. (New) Method according to claim 35, further comprising:

processing received packets according to a first set of code pages;

creating a second set of code pages;

initiating processing received packets according to said second set of code pages; and

continuing processing packets received before said initiating, according to said first set of code pages.

37. (New) Method according to claim 36, comprising within said step of creating a second set of code pages the steps of:

assigning one or more members of an existing code page set to be members of said second set of code pages; and
removing one or more code pages from said second set of code pages.

38. (New) Method according to claim 36, comprising within said step of creating a second set of code pages the steps of:

creating a new code page; and
assigning said new code page to be a member of said second set of code pages.

39. (New) Method according to claim 36, further comprising:

removing a code page from the memory when the code page is no longer a member of any set of code pages in use.

40. (New) Method according to claim 34 wherein each page of code is associated with a reference number for observing the order of the code pages.

41. (New) Method according to claim 40 wherein the order of any two code pages is determined by comparing values of $v(x)$ calculated from the reference numbers associated with the code pages, $v(x)$ being calculated substantially by the formula

$$v(x) = r(x) - r(b) \bmod M$$

where $r(x)$ is the reference number associated with a code page x being compared, $r(b)$ is the reference number of a base code page, and M is the size of a set of allowed reference numbers $(0, 1, 2, \dots, M-1)$.

42. (New) A computer readable medium having computer readable instructions to instruct a computer to perform a method comprising:

- receiving data packets for processing;
- processing the received packets according to compiled packet filtering code in a memory;
- and
- managing the compiled code in a plurality of pieces.

43. (New) Computer readable medium according to claim 42 wherein the method further comprises:

- incrementally compiling at least one packet filtering rule into least one piece of compiled code; and
- updating the memory with said at least one piece of compiled code.

44. (New) Computer readable medium according to claim 42 wherein said pieces are code pages having a predetermined length, the method further comprising:

- implementing shadow paging of code pages.

45. (New) Computer readable medium according to claim 44 wherein the method further comprises:

processing received packets according to a first set of code pages;
creating a second set of code pages;
initiating processing received packets according to said second set of code pages; and
continuing processing packets received before said initiating, according to said first set of code pages.

46. (New) Computer readable medium according to claim 42 wherein the medium further has a library of software routines for processing data packets.

47. (New) A computer readable medium having computer readable instructions to instruct a computer to perform a method according to any one of claims 31 to 41.

48. (New) A computer network node for processing data packets according to compiled filter code, comprising:

means for receiving data packets for processing;

means for processing the received packets according to compiled code in a memory; and

means for managing the compiled code in a plurality of pieces.

49. (New) Computer network node according to claim 48, further comprising:
means for incrementally compiling at least one packet processing rule and producing at least one piece of compiled code; and
means for updating the memory with said at least one piece of compiled code.

50. (New) Computer network node according to claim 48 wherein the pieces are pages of compiled code having a predetermined length, further comprising:

means for implementing shadow paging of code pages.

51. (New) Computer network node according to claim 50, further comprising:

means for processing received packets according to a first set of code pages;

means for providing a second set of code pages;

means for initiating processing received packets according to said second set of code pages; and

means for continuing processing packets received before said initiating, according to said first set of code pages.

52. (New) Computer network node according to claim 48, wherein the node is a virtual private network node.

53. (New) Computer network node according to claim 48, wherein the node is a router node.

54. (New) Computer network node according to claim 48, wherein the node is a firewall node.

55. (New) Computer network node according to claim 48, wherein the node is a workstation.

56. (New) A system for processing data packets and managing compiled packet filtering code, comprising:

means for compiling packet filtering code and writing the compiled code to a memory;

means for receiving data packets for processing;

means for processing the received packets according to the compiled code; and

means for managing the compiled filter code in a plurality of pieces.

57. (New) System according to claim 56, further comprising:

means for incrementally compiling at least one rule into at least one piece of compiled code; and

means for updating the memory with said at least one piece of compiled code.

58. (New) System according to claim 56 wherein the pieces are pages of compiled code having a predetermined length, further comprising:

means for implementing shadow paging of pages of compiled code.

59. (New) System according to claim 58, further comprising:

means for processing packets according to a first set of code pages,

means for creating a second set of code pages;

means for initiating processing received packets according to said second set of code pages; and

means for continuing processing packets received before said initiating, according to said first set of code pages.

60. (New) System according to claim 56, wherein the memory has a first access port and a second access port, the system further comprising:

means for processing data packets, said processing means being arranged to access said memory via said first access port, and said means for managing the compiled filter code being arranged to access said memory via said second access port.